

文章编号: 2095-4980(2018)04-0735-07

一种基于时延配置表的 FPGA 静态时序分析算法

喻 伟, 陈恩耀, 马海燕, 祝周荣, 宋雷军, 王永孟

(上海航天电子技术研究所, 上海 201109)

摘要: 为减小现场可编程门阵列(FPGA)关键路径的延时误差, 提出一种基于时延配置表的静态时序分析算法。算法建立了一种基于单元延时与互连线延时配置表的时延模型。该模型考虑了工艺角变化对延时参数的影响, 同时在时序分析过程中, 通过分析路径始节点与终节点的时钟关系, 实现了复杂多时钟域下的路径搜索与延时计算。实验结果表明, 与公认的基于查找表的项目评估技术(PERT)算法和VTR算法相比, 关键路径延时的相对误差平均减少了8.58%和6.32%, 而运行时间平均仅增加了19.96%和9.59%。

关键词: 现场可编程门阵列; 静态时序分析; 配置表; 关键路径

中图分类号: TN702

文献标志码: A

doi: 10.11805/TKYDA201804.0735

A FPGA static timing analysis algorithm based on delay collocation table

YU Wei, CHEN Enyao, MA Haiyan, ZHU Zhouong, SONG Leijun, WANG Yongmeng
(Shanghai Aerospace Electronic Technology Institute, Shanghai 201109, China)

Abstract: A static timing analysis algorithm is proposed, which applies the delay collocation table, to reduce the relative error of critical path delay in Field Programmable Gate Array(FPGA). Based on the collocation table model of the logic element delay and interconnect delay, the algorithm takes into account the process corner variation's effect on delay parameters. In timing analysis phase, by computing the clock relationship between source node and sink node, path searching and delay calculating in multi-clock domains are achieved. Experimental results demonstrate that the relative error of critical path delay is reduced by 8.58% and 6.32% respectively on average when compared with the Program Evaluation and Review Technique(PERT) and the VTR algorithm, while the run time is only increased by 19.96% and 9.59% respectively on average.

Keywords: Field Programmable Gate Array; static timing analysis; collocation table; critical path

随着现场可编程门阵列(FPGA)芯片的广泛应用, 时序已成为衡量 FPGA 芯片性能的一个重要因素。在大规模的 FPGA 设计中, 静态时序分析由于其较快的运行速度与无需输入向量的特性, 已成为时序验证的主要手段。FPGA 静态时序分析的目的在于特定的时序模型下, 对 FPGA 应用电路中的路径进行分析, 以判断其是否满足设计者给定的时序约束。

已有的研究表明, 传统基于项目评估技术(PERT)的静态时序分析算法^[1-3], 由于采用非线性的查找表时延模型(Non-Linear Delay Model, NLDM), 延时被表示成二维查找表的形式, 查找表仅存储了有限数量的输入信号摆率与输出信号负载等变量的组合, 并给出了这些输入变量对应的延时参数。如果查找表中没有所需的输入变量组合, 则需要根据邻近的延时进行线性插值求出。然而, 随着工艺尺度的不断缩小, 线性插值得到的延时误差越来越大, 难以满足应用场合的要求。当在 Gantt Chart 图上寻找最长及最短的时延路径时, 其关键路径结果较之于 SPICE 仿真往往不够准确; Hitchcock 在 PERT 的基础上, 提出一种基于块的分析方法(Block Oriented Techniques, BOT)^[4], 这种方法通过层次化门级网表对应的时序图并改进搜索策略, 也能找到最长时延路径, 但其时延结果较之于 SPICE 仿真通常比较悲观; David Blaauw 在 BOT 的基础上, 提出一种最迟传输模型(Latest Propagation Model, LPM)^[5], 这种模型将时序图上各逻辑门节点的多扇入信号中具有最晚到达时间的信号作为传输对象, 通过拓扑遍历, 找到最长时延路径, 但其时延结果较之于 SPICE 仿真通常又过于乐观; 清华大学的燕

收稿日期: 2017-03-12; 修回日期: 2017-07-09

基金项目: 国防基础科研计划资助项目(JCKY2016203B11)

昭然在 LPM 模型的基础上, 提出一种考虑门逻辑功能的最长路径搜索算法^[6], 它提高了逻辑门单元延时的精确度, 但由于没有考虑到互连线延时的优化, 其路径时延结果仍不够精确。以上这些基于 PERT 的算法, 除了其查找表时延模型与关键路径时延结果不够准确外, 还没有考虑到工艺角变化对器件时延性能的影响, 以及复杂多时钟域对静态时序分析的影响。

基于以上分析, 本文提出一种采用时延配置表的静态时序分析算法来解决以上问题。算法首先建立了更为精确的基本逻辑单元延时与互连通道连线单元延时的配置表, 然后在路径搜索与延时计算时, 考虑了工艺角变化给时延模型带来的影响; 提出了更为复杂的基于多时钟域的路径分析算法。本算法不仅可以有效减小关键路径延时的相对误差, 还可以适用于复杂多时钟域下的电路分析。

1 概念说明

1.1 FPGA 时序图

FPGA 的静态时序分析是在时序图上进行的, FPGA 的时序图 $G(V,E)$ 是一个有向无环图(Directed Acyclic Graph, DAG)。在时序图中, 每个被使用到的输入模块引脚、输出模块引脚、基本逻辑单元(Logic Element, LE)引脚构成了一个节点 V 。每对输入模块引脚与基本逻辑单元引脚、基本逻辑单元引脚与基本逻辑单元引脚、基本逻辑单元引脚与输出模块引脚之间的连接构成了时序图上的一个边 E 。输入模块没有输入边, 这需要生成一个虚拟始点(INPAD_SOURCE); 输出模块没有输出边, 也需要生成一个虚拟终点(OPAD_SINK)。对于输入模块, 输出引脚的信号从芯片中来, 并传输到基本逻辑单元或输出模块; 对于输出模块, 输出引脚的信号从逻辑块或输入模块中来, 并传输到外部芯片, 如图 1 所示。需要注意的是, 如果 LE 中的寄存器工作在组合模式, 只需把该 LE 的输入引脚连向输出引脚; 如果 LE 中的寄存器工作在时序模式, 则需生成 2 个新节点, 一个是寄存器虚拟始节点(FF_SOURCE), 用于连向 LE 输出引脚, 这意味着寄存器在时钟到达后才能产生输出; 另一个是寄存器虚拟终节点(FF_SINK), 用于接收 LE 输入引脚传递来的信号, 没有输出边。

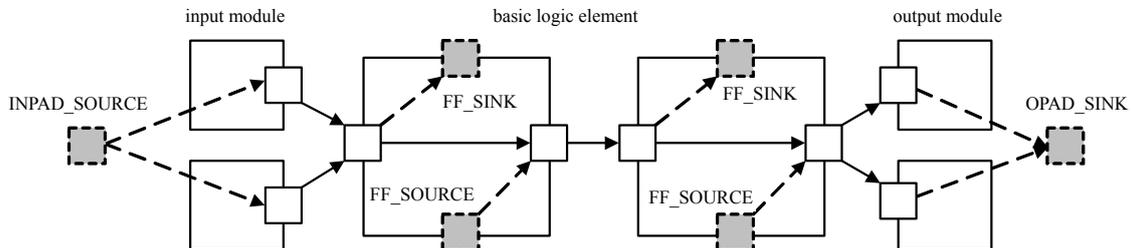


Fig.1 Diagram of FPGA timing graph
图 1 FPGA 的时序图示例

FPGA 时序图的构造过程如下: 首先扫描网表信息, 得到 LE 上输出引脚的扇出边个数, 并为 LE 上的每个引脚构建时序图上相应点的映射关系; 然后为网表中输入模块、输出模块、基本逻辑单元等建立节点信息和连接边信息; 接着层次化时序图, 得到图中每一层节点的链表; 最后再检查时序图的节点与边是否符合网表要求。

下面给出本文使用的几个概念的定义:

定义 1: 在时序图 $G(V,E)$ 中, 对任意一对顶点 v 和 u , 如果存在从 v 到 u 的边 $(v,u) \in E$, 那么 v 称作 u 的扇入 (fan-in), u 称作 v 的扇出 (fan-out), $D(v,u)$ 表示从 v 到 u 的延时。

定义 2: 在时序图 $G(V,E)$ 中, 对任意一个节点 v , $T_{\text{arrival}}[v]$ 表示信号从路径始节点到 v 的实际到达时间, $T_{\text{required}}[v]$ 表示从路径始节点到 v 的要求到达时间。

1.2 FPGA 关键路径

FPGA 的关键路径是指时序图中源节点与目标节点之间的传输路径延时最长的路径。每次布线迭代完后, 时序分析的过程分成对时序图的 2 次遍历。第一遍在时序图上进行自上而下的顺向搜索, 求出每个节点的实际到达时间 $T_{\text{arrival}}(i)$; 第二遍进行自下而上的逆向搜索, 求出每个节点的要求到达时间 $T_{\text{required}}(i)$, 然后再计算出每个节点的 $Slack$ 值, 如式(1)~式(3):

$$T_{\text{arrival}}(i) = \text{Max}_{v_j \in \text{fan-in}(i)} \{T_{\text{arrival}}(j) + D(j, i)\} \quad (1)$$

$$T_{\text{required}}(i) = \text{Min}_{v_j \in \text{fan-out}(i)} \{T_{\text{required}}(j) - D(i, j)\} \quad (2)$$

$$Slack(i) = T_{required}(i) - T_{arrival}(i) \tag{3}$$

若节点 i 的 $Slack$ 值为 0, 则可以确定此节点就是组成关键路径的节点, 进而可以找到一条完整的关键路径。

2 FPGA 延时计算

在已有文献资料中, 研究人员采用的延时模型普遍为 NLDM 查找表时延模型^[7]与 Elmore 模型^[8]。对于 NLDM 查找表时延模型, 逻辑门延时是输入信号摆率与输出负载电容的函数。由于使用了内插方程, NLDM 的估值通常偏离正确值, 这是因为内插方程峰值取决于负载与斜率变量的耦合系数, 小的误差也可能导致计算结果与仿真结果之间产生很大的偏差。另外, 为解决工艺的可变性问题, 必须维护大量的 NLDM 库; 为产生 NLDM 表, 也必须有一个输入斜率与负载的矩形栅格, 从而增加了设计的复杂度。对于互连线 Elmore 时延模型, 互连线延时不仅取决于布线资源节点所生成电阻电容树的形状, 还受到各节点累计电阻、电容及缓冲器的影响。不同的金属电阻、金属电容、寄生电阻和寄生电容将导致不同的互连线延时, 从而使得连线单元的时序不稳定。

因此, 为了克服上述模型的缺点, 保证 FPGA 时序性能的稳定和准确, 本文从算法分析的角度将 FPGA 的路径延时分为 2 部分: 单元延时与互连延时。

2.1 FPGA 单元延时

FPGA 的单元延时是指 LE 单元的引脚至引脚延时。LE 是 FPGA 中最小的功能电路单位, 通常包含一个查找表(Look-Up Table, LUT)和一个由时钟控制的触发器等^[9], 可以实现简单的组合或时序逻辑电路功能。图 2 给出了一个考虑时序图虚拟节点的 FPGA 基本逻辑单元结构示意图, 其中, FF_SINK 为时序图中寄存器的虚拟终节点, 表示寄存器的虚拟输出端(注意与寄存器输出端 Q 不同), 没有扇出; FF_SOURCE 为时序图中寄存器的虚拟始节点, 表示寄存器的寄存器虚拟输入端(注意与寄存器输入端 D 不同), 没有扇入; k 为 LUT 表输入端的个数(k 取 4)。

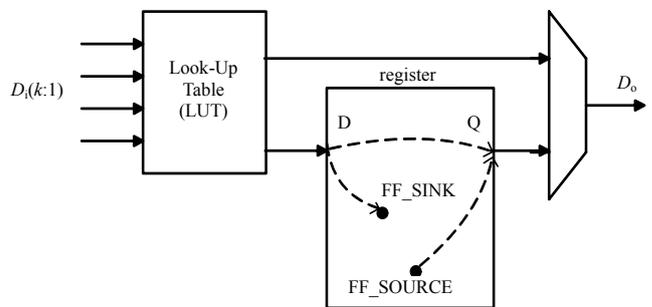


Fig.2 Diagram of FPGA Basic Logic Element with timing graph virtual nodes

图 2 考虑时序图虚拟节点的 FPGA 基本逻辑单元结构示意图

当寄存器工作在组合模式时, 对于同一寄存器, LE 不同输入引脚节点到寄存器输出引脚节点的延时有差别, 表明不同 LUT 输入端(a, b, c, d)到寄存器输出端 Q 的组合延时是不同的。

当寄存器工作在时序模式时, 对于同一寄存器, LE 不同输入引脚节点到寄存器虚拟终节点的延时有差别, 表明不同 LUT 输入端(a, b, c, d)到寄存器虚拟输出端 FF_SINK 的时序延时也是不同的。同时, 为了评估寄存器虚拟始节点到 LE 输出引脚节点的延时, 也需要计算出寄存器虚拟输入端 FF_SOURCE 到寄存器输出端 Q 的时序延时。不同的输入输出情况对应不同的单元延时, 将这些不同类型的单元延时归纳在一个配置表中, 以方便布线与时序分析, 如表 1 所示。

表 1 FPGA 单元延时配置表

cell delay mode	to register FF_SOURCE	to register Q
LE a input	sequential delay	combinational delay
LE b input	sequential delay	combinational delay
LE c input	sequential delay	combinational delay
LE d input	sequential delay	combinational delay
	register FF_SOURCE	sequential delay

2.2 FPGA 互连延时

FPGA 的互连延时是指信号经过互连通道内连线单元的延时。连线通道由一组规则的连线单元组成, 它具体实现逻辑阵列块之间的连接, 连线单元的长度对于系统的延时特性有很大影响。

不同的连线单元长度分布对应不同的延时, 中科院电子所的刘洋等采用多驱动型互连结构, 利用 VPR 平台, 从优化面积延时积的角度提出一种互连分布^[10-11]: 根据不同的单元长度将连线单元分为: 8 线长(L_4)、4 线长(L_3)、2 线长(L_2)及 1 线长(L_1); 根据信号在 SB 的跳

表 2 FPGA 互连线延时配置表

interconnect delay mode	horizontal channel	vertical channel
L_1	clockwise/anticlockwise delay	clockwise/anticlockwise delay
L_2	clockwise/anticlockwise delay	clockwise/anticlockwise delay
L_3	clockwise/anticlockwise delay	clockwise/anticlockwise delay
L_4	clockwise/anticlockwise/jumping delay	clockwise/anticlockwise/jumping delay

转方向又将连线单元分为: 顺时针水平、顺时针垂直、逆时针水平、逆时针垂直、jump 水平及 jump 垂直等。连线单元的数目、方向、长度、组成比例和分布, 影响了 FPGA 的布通率与资源利用率, 也直接决定了电路的信号

质量与延时特性。将这些不同类型的互连延时归纳在一个配置表中,以方便布线与时序分析,如表2所示。

2.3 本文延时建模

传统的静态时序分析方法,是完全基于对某一固定工艺角下的开关参数进行延时测试而进行的,然后假定由此产生的延时参数适用于所有设计。这种方法对 $0.13\ \mu\text{m}$ 及其以上工艺大体适用,但至深亚微米工艺,工艺角的变化会导致先前为定值的片内延时产生波动,因此会产生一个延时不稳定的情况,从而为基于“最好/最坏情况”的静态时序分析引入不确定性^[12-13]。因此,如何对工艺角变化带来的延时不确定性进行建模,并进行定性的延时计算也成为当前静态时序分析的一大难题。

已有的文献表明^[14-15],对于一定数量的芯片,受工艺角变化的影响,其单元延时与互连延时大致呈高斯分布,如图3所示。

为模拟不同工艺角对 FPGA 单元延时与互连延时的影响,本文考虑3种典型的工艺角实例,以 PMOS-NMOS 晶体管为例。在标准(Typical-Typical, TT)工艺角下,电子在 PMOS,NMOS 中传输速度正常,门器件或互连延时是一个特定值;在最好(Fast-Fast, FF)工艺角下,电子在 PMOS,NMOS 中传输速度都较快,门器件或互连延时相对 TT 较小;在最坏(Fast-Slow, FS)工艺角下,电子在 PMOS 中传输较快,而在 NMOS 中传输较慢,门器件或互连延时则相对 TT 较大。

实际设计中, FPGA 通常选择标准工艺角下的工艺角。然而,由于工艺偏差(工艺角变化)的影响, FPGA 的片内延时将不再是一个确定的量值。偏差影响下的 FPGA 单元延时与互连延时会在标准工艺角对应延时的上下摆动,并呈高斯分布,如图3所示,图中实线对应 FPGA 的单元延时分布, $\mu_1, (\mu_1 + \sigma_1)$ 及 $(\mu_1 - \sigma_1)$ 分别为其均值和上下延时边界;虚线对应 FPGA 的互连延时分布, $\mu_2, (\mu_2 + \sigma_2)$ 及 $(\mu_2 - \sigma_2)$ 也对应其均值和上下延时边界。为保证 FPGA 实际工作时的单元延时与互连延时能够最大限度地贴近于标准工艺角下的延时值,本文提出一种加权的延时模型,以保证 FPGA 片内延时的鲁棒性。

在延时模型的构建过程中,为不失普遍性,取1份 FF 工艺角下的延时样本,1份 FS 工艺角下的样本和 k 份 TT 工艺角下的延时样本。令 f 表示 FF 工艺角下的单元延时或互连线延时,为乐观估计; s 表示 FS 工艺角下的单元延时或互连线延时,为悲观估计; t 表示 TT 工艺角下的单元延时或互连线延时,为正常估计; k 为采样的 TT 工艺角的数量。

考虑一个加权的延时均值表达式:

$$\mu = (f + kt + s) / (k + 2) \quad (4)$$

观察可知,式(4)对应单元/互连延时标准差 $\sigma = (s - f) / (k + 2)$ 。当 $k = 4$ 时,均值 $\mu = (f + 4t + s) / 6$,标准差 $\sigma = (s - f) / 6$,这表明整个延时取值区间被分成6等份,且正负3倍标准差范围的取值基本可以覆盖整个延时分布。由于实际仿真时,TT 工艺角下的延时约等于 FF,FS 工艺角下延时的算术平均值,即 $t \approx (f + s) / 2$,可知 $\mu + 3\sigma \approx s$, $\mu - 3\sigma \approx f$ 。这表明悲观估计下的单元延时或互连延时 s 等效于正态分布的右边界值 $\mu + 3\sigma$,乐观估计下单元延时及互连延时 f 等效于正态分布的左边界值 $\mu - 3\sigma$,这正好解释了本章基于加权的延时均值和方差表达式在理论上符合高斯分布 $N(\mu, \sigma)$ 。式(4)的物理意义在于:通过构建一个加权的延时模型,提高了定时分析的准确度,方便了静态时序的确定性分析。

实际分析时,通过测试3种工艺角下 FPGA 各单元延时和互连线延时的值,再经过上述公式的计算,就可以得到一个加权的单元延时均值 μ_1 与互连延时均值 μ_2 ,将这些加权后的延时均值存储到一个配置表中,通过在布线中反标时序图,就可以计算各条路径的路径延时。

3 算法实现

传统基于 PERT 的静态时序分析算法并没有解决复杂多时钟域下的路径搜索与延时计算。基于此,本文提出一种基于时延配置表的静态时序分析算法,算法在时序分析过程中,考虑了节点的扇入扇出关系,通过分析路径起始节点(SOURCE)和终节点(SINK)的时钟域关系,计算各节点的最大到达时间、最小到达时间及要求到达时间,

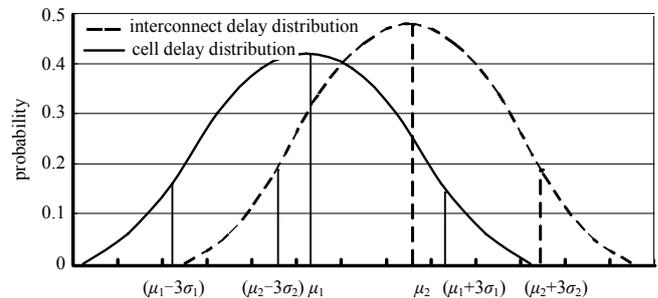


Fig.3 Diagram of FPGA cell/interconnect delay distribution with metabolic process corners

图3 考虑工艺角变化的FPGA单元延时与互连延时分布

从而解决了多时钟域下的路径搜索与延时计算问题。

具体来说,在建立完时序图后,针对某一固定的 SOURCE-SINK 对(路径始节点-路径终节点),需要分别找到最长延时路径与最短延时路径,使其始于 SOURCE 节点,终于 SINK 节点。从始节点 v 到终节点 n 的最长延时可用于进行寄存器建立时间分析,其值为 $T_{\text{arrival-max}}[n]$;从始节点 v 到终节点 n 的最短延时可用于进行寄存器保持时间分析,其值为 $T_{\text{arrival-min}}[n]$ 。在获取了一条 SOURCE-SINK 路径上所有节点的实际到达时间后,将其保存在相应的数据结构中,就可以方便后续的建立保持时间分析。

在进行建立保持时间分析,尤其是在计算节点 n 的要求到达时间时,需要考虑 v 与 n 的不同时钟关系。为方便分析,令 $T_{\text{req-setup}}[n]$ 为节点 n 用于建立时间分析的要求到达时间, $T_{\text{req-hold}}[n]$ 为节点 n 用于保持时间分析的要求到达时间。

1) v 和 n 来自同一个时钟源

若 v 和 n 对应时钟都没有被反相或者都被反相,则节点 n 的要求到达时间为:

$$\begin{cases} T_{\text{req-setup}}[n] = T_{\text{cp}} + S - T_{\text{su}} \\ T_{\text{req-hold}}[n] = 0 \end{cases} \quad (5)$$

若 v 和 n 对应时钟只有一个被反相,则节点 n 的要求到达时间为:

$$\begin{cases} T_{\text{req-setup}}[n] = T_{\text{cp}} / 2 + S - T_{\text{su}} \\ T_{\text{req-hold}}[n] = 0 \end{cases} \quad (6)$$

式中: T_{cp} 为该时钟源的周期; S 为时钟源到达 v 和 n 的时钟偏斜; T_{su} 为寄存器的内部建立时间。

2) v 和 n 来自不同的时钟源

若 v 对应时钟 clk1 , n 对应时钟 clk2 都没被反相或者都被反相,则节点 n 建立时间分析的要求到达时间为:

$$T_{\text{req-setup}}[n] = (M_s - 1)T_{\text{cp2}} + F_2 - F_1 + S - T_{\text{su}} \quad (7)$$

若 v 对应时钟 clk1 , n 对应时钟 clk2 只有一个被反相,则节点 n 的建立时间分析的要求到达时间为:

$$T_{\text{req-setup}}[n] = \frac{(M_s - 1)T_{\text{cp2}} + F_2 - F_1}{2} + S - T_{\text{su}} \quad (8)$$

令 T_{cp1} 为 clk1 的周期, T_{cp2} 为 clk2 的周期。

若 $T_{\text{cp1}} = T_{\text{cp2}}$, 则节点 n 的保持时间分析的要求到达时间为:

$$T_{\text{req-hold}}[n] = [F_2 + (M_s - 1)T_{\text{cp2}}] - M_h \cdot T_{\text{cp2}} - F_1 + S \quad (9)$$

若 $T_{\text{cp1}} \neq T_{\text{cp2}}$, 则节点 n 的保持时间分析的要求到达时间为:

$$\begin{cases} T_{\text{req-hold1}}[n] = (M_s - M_h - 1)T_{\text{cp2}} + F_2 - F_1 + S \\ T_{\text{req-hold2}}[n] = (M_s - M_h)T_{\text{cp2}} - T_{\text{cp1}} + F_2 - F_1 + S \\ T_{\text{req-hold}}[n] = \max(T_{\text{req-hold1}}[n], T_{\text{req-hold2}}[n]) \end{cases} \quad (10)$$

式(7)~式(10)中: F_1 为 clk1 的偏置; F_2 为 clk2 的偏置; M_s 为建立时间分析的多周期约束条件,用来约束终点寄存器的建立时间数据采样周期; M_h 为保持时间分析的多周期约束条件,用来约束终点寄存器的保持时间数据采样周期。

最终,在计算完要求到达时间和实际到达时间并建立保持时间分析时,各条路径的时序裕量就可以表示为:

$$\begin{cases} Slack_{\text{setup}} = T_{\text{req-setup}}(n) - T_{\text{arrival-max}}(n) \\ Slack_{\text{hold}} = T_{\text{arrival-min}}(n) - T_{\text{req-hold}}(n) \end{cases} \quad (11)$$

4 实验结果

实验采用的 FPGA 芯片结构为:每个逻辑块包含 8 个 4 输入的基本逻辑单元,互连通道采用不同比例的 2 线长、4 线长及 8 线长的双向连接线,连接开关为全连接,交叉开关采用 subset 结构。实验的硬件平台为:CPU 3.6 GHz,内存 1.0 GB。

在试验比较对象的选择上,以经典的 PERT 算法和最新的加拿大多伦多大学开发的 VTR 算法^[16]作为本章算法的比较对象。基于中科院电子所自主研发的国产 FPGA 应用软件平台 Passkey,本节对 8 个测试电路首先进行装箱、布局布线,然后分别用 SPICE 工具、本章算法、PERT 算法及基于 PERT 的 VTR 算法进行时序分析,重点比较了关键路径延时及运行时间等指标。

4.1 关键路径延时

对于 PERT 算法和 VTR 算法,实验采用基于 TT 工艺的逻辑门查找表延时模型与互连线 Elmore 延时模型;本文算法采用基于加权配置表的延时模型。

为评估本文算法获取关键路径延时的精确性,表 3 给出了 8 个电路分别在 PERT 算法、VTR 算法、本文算法及标准工艺 SPICE 仿真条件下的关键路径延时结果。从表中可以看出,与 SPICE 结果相比,PERT 算法的平均相对误差达到了 11.72%,VTR 算法的平均相对误差达到了 9.46%,而本文算法的平均相对误差仅为 3.14%,比 PERT 算法减少了 8.58%,比 VTR 算法减少了 6.32%,表明本文基于加权配置表的延时模型具有更好的准确性。

表 3 关键路径延时
Table3 Critical path delay

circuits	circuit description		critical path delay/ns				relative deviation/%		
	function description	block number	SPICE	proposed algorithm	PERT	VTR	proposed algorithm	PERT	VTR
SignalTap	embedded logic analyzer	431	8.329	8.126	12.176	11.567	2.44	46.19	38.87
Counter16	16 bit counter	3	8.346	8.079	7.332	7.698	3.20	12.15	7.76
Mul32	32 bit multipliers	13	9.400	9.123	9.129	9.174	2.95	2.88	2.40
Shift2500	2 500 bit shift registers	250	11.202	10.793	10.620	10.673	3.65	5.20	4.72
Drfm	digital radio frequency memory	72	12.232	11.784	12.076	12.136	3.66	1.28	1.27
GateCount	XOR chain circuit	1 092	9.998	9.670	8.668	9.100	3.28	13.30	8.98
Tmr	triple modular redundancy	22	13.715	13.459	14.169	13.530	1.87	3.31	2.69
DspMac	digital signal processing	8	14.126	13.548	12.796	12.859	4.09	9.42	8.96
mean value							3.14	11.72	9.46

4.2 运行时间

由于本文算法是在延时配置表的基础上进行路径搜索,节省了 PERT 算法与 VTR 算法中互连线 Elmore 延时的电阻电容树拓扑遍历时间。与此同时,由于本文算法在单元延时与互连延时的建模过程中也耗费了一部分建模时间,这部分建模时间也应该包含在整体运行时间中。表 4 列出了 2 种算法运行时间的比较结果,从表中可以看出,与 PERT 算法和 VTR 算法相比,本文算法的运行时间从 Mul32 电路分别增加了 9.11%,3.92%到 GateCount 电路分别增加了 41.81%,23.31%,整体来看,算法运行时间平均分别增加了 19.96%,9.59%。

表 4 运行时间比较
Table4 Comparison of run time

circuits	PERT run time/s	VTR run time/s	proposed algorithm				
			search time/s	modeling time/s	total time/s	increase compared with PERT/%	increase compared with VTR/%
SignalTap	0.644	0.708	0.250	0.487	0.737	14.44	4.10
Counter16	1.235	1.296	0.062	1.305	1.367	10.69	5.48
Mul32	2.601	2.731	0.203	2.635	2.838	9.11	3.92
Shift2500	23.22	24.38	1.469	26.70	28.17	21.33	15.55
Drfm	3.887	4.278	0.812	3.748	4.560	17.31	6.59
GateCount	109.20	125.60	41.29	113.60	154.90	41.81	23.31
Tmr	4.311	4.740	0.656	4.453	5.109	18.51	7.78
DspMac	8.784	10.100	0.140	10.970	11.110	26.50	10.01
mean	—	—	—	—	—	19.96	9.59

5 结论

本文提出一种采用延时配置表的静态时序分析算法。首先,根据 FPGA 的基本逻辑单元的工作模式与互连通道内连线单元的分布情况,将单元延时与互连线延时进行分类,并建立一种考虑工艺变化的配置表延时模型。其次,提出一种路径分析算法,通过分析多时钟域下各个路径始节点与终节点的时钟关系,实现了复杂多时钟域下的路径搜索与延时计算。实验结果表明,与 PERT 算法和 VTR 算法相比,本文算法可有效减小关键路径延时的相对误差。

由于本文算法是在具体的 FPGA 芯片结构上实现的,主要的比较对象为学术开源代码,缺乏与商业 EDA 时序分析工具的比较,算法的不足之处主要是通用性与移植性不够好,下一步工作将尝试在其他的 EDA 平台中应用本文算法。

参考文献:

- [1] KIM S D,HAMMOND R K,BICKEL J E. Improved mean and variance estimating formulas for PERT analysis[J]. IEEE Transactions on Engineering Management, 2014:61(2):362-369.
- [2] BENICK J,STEINHAUSER B,MULLER R,et al. High efficiency n-type PERT and PERL solar cells[C]// Proceedings of IEEE 40th Photovoltaic Specialist Conference(PVSC). Denver,CO,USA:IEEE, 2014:3637-3640.
- [3] KIRKPATRICK T I,CLARK N R. PERT as an aid to logic design[J]. IBM Journal of Research and Development, 1966,10(2): 135-141.
- [4] HITCHCOCK R B. Timing verification and the timing analysis program[C]// Proceedings of the 19th Design Automation Conference. Las Vegas,NV,USA:IEEE, 1982:594-604.
- [5] BLAAUW D,ZOLOTOV V,SUNDARESWARAN S,et al. Slope propagation in static timing analysis[C]// Proceedings of the 17th International Conference on Computer Aided Design. San Jose,CA,USA: IEEE, 2000:338-343.
- [6] YAN Zhaoran,YANG Huazhong,LOU Rong,et al. The longest path search algorithm considering gate logic functions[J]. Journal of Microelectronics, 2004,34(3):285-288.
- [7] KAYSSI A I,SAKALLAH K A,MUDGE T N. The impact of signal transition time on path delay computation[J]. IEEE Trans. on Circuits and Systems-II:Analog and Digital Signal Processing, 1993,40(5):302-309.
- [8] BOESE K,KAHNG A,MCCOY B,et al. Fidelity and near optimality of Elmore-based routing constructions[C]// Proceedings of the 6th International Conference on Computer Design:VLSI in Computers and Processors. Cambridge,MA,USA:IEEE, 1993:81-84.
- [9] KOULOBERIS J L,GAMA A E. FPGA performance versus cell granularity[C]// Proceedings of the 13th Custom Integrated Circuits Conference. San Diego,CA,USA:IEEE, 1991:6.2/1-6.2/4
- [10] YU Wei,YANG Haigang,LIU Yang,et al. Timing slack optimization approach using FPGA hybrid routing strategy of rip-up-retry and pathfinder[J]. Journal of Electronics(China), 2014:31(3):246-255.
- [11] LIU Yang,YANG Haigang,YU Wei,et al. An FPGA timing routing algorithm based on pathfinder and rip-up and retry approach[J]. Journal of Computer-Aided Design & Computer Graphics, 2014,26(1):138-145.
- [12] MOHANTY S P,KOUGIANOS E. Incorporating manufacturing process variation awareness in fast design optimization of nanoscale CMOS VCOs[J]. IEEE Transactions on Semiconductor Manufacturing, 2014,27(1):22-31.
- [13] LAO Y J,PARHI K K. Statistical analysis of MUX-based physical unclonable functions[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2014,33(5):649-662.
- [14] TANG Q,RODRIGUEZ J,ZJAJO A,et al. Statistical transistor-level timing analysis using a direct random differential equation solver[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2014,33(2):210-223.
- [15] TANG Q,ZJAJO A,BERKELAARAND M,et al. Considering crosstalk effects in statistical timing analysis[J]. IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, 2014,33(2):318-322.
- [16] ROSE J,LUU J,YU C W,et al. The VTR project: architecture and CAD for FPGAs from Verilog to routing[C]// Proceedings of ACM International Symposium on Field-Programmable Gate Arrays. Monterey,CA,USA:IEEE, 2012:77-86.

作者简介:



喻 伟(1986-),男,河南省商水县人,博士,主要研究方向为集成电路计算机辅助设计及 FPGA 验证技术.email:yuwei19861119@126.com.

陈恩耀(1990-),男,安徽省芜湖市人,硕士,主要研究方向为 FPGA 验证.

马海燕(1986-),女,江苏省南通市人,学士,主要研究方向为软件测评.

宋雷军(1980-),男,上海市人,高级工程师,主要研究方向为软件开发.

王永孟(1978-),男,上海市人,硕士,高级工程师,主要研究方向为软件测评.

祝周荣(1977-),女,上海市人,硕士,高级工程师,主要研究方向为 FPGA 验证.